Swing/SWT Integration with the Albireo Project

Gordon Hirsch (SAS)
Bruno Haible (ILOG)
Why?

- Why embed Swing in RCP/SWT applications?
  - Reuse existing Swing GUIs
    - Incremental adoption of Eclipse technologies
    - Use vendor libraries built upon Swing

- Why the need for Albireo?
  - SWT’s SWT_AWT bridge is low-level
  - Albireo is much easier to use
    - Common integration tasks are performed
    - Some complexity from multithreading is hidden
Features

• Synchronized look&feel
• Reduced flicker of embedded Swing components
• Correct behavior of modal dialogs across both toolkits
• Proper focus handling, tab traversal
• Easier multithreading
• Smooth integration of size/layout management
• SWT pop-up menus on Swing components
Code Example

```java
SwingControl control =
    new SwingControl(parent, SWT.NONE) {
        protected JComponent createSwingComponent() {
            return new JMagicPanel();
        }
        public Composite getLayoutAncestor() {
            return parent;
        }
    };
```
Status

• Synchronized look&feel
• Flicker
• Modal dialogs
• Focus handling
• Keystroke contention
• Multithreading
• Size/layout management
• SWT pop-up menus
More about Albireo

• Visit our BOF!
  Wednesday, 8:45 pm, Room 209/210
  Your feedback will be welcome.

• Project home page: http://www.eclipse.org/albireo

• Project wiki:
  http://wiki.eclipse.org/Albireo_Project